

*In the Claims*

The status of claims in the case is as follows:

1     1.   [Previously presented]   A cache coherency system for a  
2     shared memory parallel processing system including a  
3     plurality of processing nodes, comprising:

D2 4     a single multi-stage communication network for  
5     interconnecting said processing nodes, said network  
6     including a dual priority switch at each node for  
7     selectively operating in normal low priority mode and  
8     camp-on high priority mode;

9  
10     each said processing node including a unique section of  
11     shared memory which is not a cache;

12     each said processing node including one or more caches  
13     for storing a plurality of cache lines;

14     a cache coherency directory which is distributed to  
15     each of said nodes for tracking which of one or more of  
16     said nodes have copies of each cache line; and

17 an adapter for storing changed data immediately to said  
18 unique section of shared memory regardless of which of  
19 said nodes is changing the data and which of said nodes  
20 includes the section of shared memory to be changed,  
21 such that said shared memory always contains the most  
22 recent data according to a two hop process including in  
23 hop 1) a requesting node requests most recent data of a  
24 home node, and in hop 2) said home node immediately  
25 returns said most recent data from its shared memory to  
26 said requesting node.

27 2. [Withdrawn] A shared memory parallel processing  
28 system including a plurality of processing nodes,  
29 comprising:

30 a multi-stage communication network for interconnecting  
31 said processing nodes, said network including a  
32 plurality of self-routing switches cascaded into first,  
33 middle and last stages, each said switch including a  
34 plurality of switch inputs and a plurality of switch  
35 outputs, each of said switch outputs of each said  
36 switch coupled to a different switch input of others of  
37 said switches, switch outputs of said last stage  
38 switches including network output ports, and switch

39 inputs of said first stage switches comprising network  
40 input ports;

41 each processing node including:

42 - a network adapter for transmitting and receiving  
43 messages with respect to other processing nodes  
44 over said network;

45 a local processor;

46 at least one private write-through cache;

47 a section of shared memory organized into a  
48 plurality of cache lines, each cache line  
49 including one or more addressable memory  
50 locations;

51 a cache coherency directory for tracking which of  
52 said nodes have copies of each cache line;

53 said local processor at a first processing node being  
54 operable for writing data to said private cache at said  
55 first node, as the same data is written to either

56 shared memory at said first node or sent over said  
57 network for writing to the shared memory and private  
58 cache of a second processing node.

1 3. [Withdrawn] The shared memory parallel processing  
2 system of claim 2, wherein said section of shared memory is  
3 divided into first and second portions, said first portion  
4 for storing unchangeable data, and said second portion for  
5 storing changeable data.

1 4. [Withdrawn] The shared memory parallel processing  
2 system of claim 3, said cache coherency directory for this  
3 processing node listing which nodes of the plurality of  
4 nodes have accessed copies of said cache lines in said  
5 second portion of shared memory at this processing node.

1 5. [Withdrawn] The shared memory parallel processing  
2 system of claim 4, wherein each said processing node is  
3 operable for reading, storing, and invalidating the shared  
4 memory at any of said plurality of processing nodes  
5 selectively by transmitting and receiving messages over said  
6 network, a first message type for requesting the read of a  
7 cache line, a second message type for returning the

8 requested cache line, a third message type for storing a  
9 cache line, and a fourth message type for invalidating a  
10 cache line.

1 6. [Withdrawn] The shared memory parallel processing  
2 system of claim 5, said network adapter further comprising:

3 a first buffer for transmitting to said network shared  
4 memory read command messages of said first message type  
5 and said second message type;

6 a second buffer for transmitting to said network shared  
7 memory store command messages of said third message  
8 type;

9 a third buffer for transmitting to said network  
10 invalidate messages for said cache coherency directory  
11 of said fourth message type;

12 a fourth buffer for receiving from said network shared  
13 memory read command messages of said first message type  
14 and said second message type;

15 a fifth buffer for receiving from said network shared

16 memory store command messages of said third message  
17 type; and  
  
18 a sixth buffer for receiving from said network  
19 invalidate messages for said cache coherency directory  
20 of said fourth message type.

1 7. [Withdrawn] A shared memory parallel processing  
2 system, comprising:  
3 a plurality of nodes, each node including a node  
4 memory, at least one cache, and a memory controller;  
5 a multi-stage switching network for  
6 interconnecting said processing nodes, said switching  
7 network including a plurality of self-routing switches  
8 cascaded into first, middle and last stages, each said  
9 switch including a plurality of switch inputs and a  
10 plurality of switch outputs, each of said switch  
11 outputs of each said switch coupled to a different  
12 switch input of others of said switches, switch outputs  
13 of said last stage switches including network output  
14 ports, and switch inputs of said first stage switches  
15 comprising network input ports;  
16 a system memory distributed to said node memories  
17 of said plurality of nodes and accessible by any node;

18 each said node memory being organized into a plurality  
19 of addressable word locations;  
20 said memory controller at this node operable for  
21 performing local memory access to the portion of system  
22 memory at this node and for performing remote memory  
23 access over said network to the portion of system  
24 memory at other nodes; and  
25 a cache coherency controller at this node being  
26 responsive to both local memory accesses and remote  
27 memory accesses to data stored in a word location of  
28 said node memory at this node for caching accessed data  
29 in the cache of this node and for communicating data  
30 for assuring cache coherency throughout said system  
31 over said network.

1 8. [Withdrawn] The shared memory processing system of  
2 claim 7, said system memory being distributed in equal  
3 portions to each said node memory; and said node memory  
4 being further sub-divided into a first memory section for  
5 storing data that is changeable and a second memory section  
6 for storing data that is unchangeable.

1 9. [Withdrawn] The shared memory processing system of

2 claim 7, further comprising node indicia for uniquely  
3 identifying each node.

1 10. [Withdrawn] The shared memory processing system of  
2 claim 7, said cache coherency controller further comprising:

3 an invalidation directory for storing a list of node  
4 indicia identifying those nodes having accessed a copy  
5 of each said cache line of node memory since the last  
6 time the cache line was changed.

1 11. [Withdrawn] The shared memory processing system of  
2 claim 10, said cache coherency controller further  
3 comprising:

4 an overflow directory for expanding said invalidation  
5 directory when the list of node indicia for a cache  
6 line becomes too long to contain entirely with said  
7 invalidation directory.

1 12. [Withdrawn] A shared memory parallel processing  
2 system, comprising:



3 a plurality of nodes, each node including a node  
4 memory, at least one cache, and a memory controller;

5 a multi-stage switching network for interconnecting  
6 said processing nodes, said switching network including  
7 a plurality of self-routing switches cascaded into  
8 first, middle and last stages, each said switch  
9 including a plurality of switch inputs and a plurality  
10 of switch outputs, each of said switch outputs of each  
11 said switch coupled to a different switch input of  
12 others of said switches, switch outputs of said last  
13 stage switches including network output ports, and  
14 switch inputs of said first stage switches comprising  
15 network input ports; and

16 a network adapter responsive to a node connection  
17 request for establishing a connection path to a target  
18 node, first by attempting to establish a quick  
19 connection path across a plurality of segments of said  
20 switching network to said target node, and upon  
21 determining any one of said plurality of segments is  
22 not available, issuing a camp-on connection request to  
23 said target node.

1 13. [Withdrawn] The shared memory parallel processing  
2 system of claim 12, further comprising:

3 said plurality of nodes each coupled to one of the  
4 network output ports and to one of the network input  
5 ports;

6 each node further including:

7 receive means for receiving a data message; and

8 send means for sending a data message across an  
9 n-stage switching network from a local node to a  
10 remote node, said send means generating said  
11 connection request including n sequential  
12 connection commands, each sequential connection  
13 command selecting one of said plurality of  
14 connection segments for each of the n switch  
15 stages of said network.

1 14. [Withdrawn] The shared memory parallel processing  
2 system of claim 12, each said switch being responsive to  
3 node connection requests and camp-on connection requests for  
4 establishing connection segments from any switch input port

5 to any switch output ports.

1 15. [Withdrawn] The shared memory parallel processing  
2 system of claim 14, each said switch further comprising:

3 a data bus for transferring said data message;

4 a rejection control line for signalling back to a  
5 sending node a rejection of any connection request;

6 an acceptance control line for signalling back to said  
7 sending node the acceptance of a camp-on connection  
8 request;

9 a valid control line for receiving from said sending  
10 node the activation of a node connection request; and

11 a camp-on control line for receiving from said sending  
12 node the activation of a camp-on connection request.

1 16. [Withdrawn] A bi-directional network adapter for  
2 interfacing a local node of a shared memory parallel  
3 processing system to a multi-stage switching network for  
4 communications with a remote node, each said node including

5 a node memory including a changeable portion and an  
6 unchangeable portion, and a node cache; said network adapter  
7 comprising:

8 a plurality of send buffers for storing and forwarding  
9 data messages from said local node to said remote node  
10 over said network, and

11 a plurality of receive buffers for storing and  
12 forwarding a plurality of data messages from said  
13 remote node to said local node over said multi-stage  
14 network;

15 said data messages including:

16 an invalidation message for invalidating a cache  
17 line that was accessed by a remote node after said  
18 cache line has changed;

19 a read request message for requesting access of a  
20 cache line from a remote node;

21 a response message for returning a cache line over  
22 the network to a remote node that has previously

23 requested data by a read request message; and  
24 a store message storing a changed cache line to a  
25 remote node.

1 17. [Withdrawn] The network adapter of claim 16, said  
2 data messages further including a message header comprising:

3 message type differentiation indicia;

4 destination node indicia for identifying a node for  
5 receiving said data message over said network;

6 source node indicia for identifying a node for  
7 transmitting said data message over said network;

8 message length indicia for defining the variable number  
9 of words included in said data message;

10 memory area indicia for defining whether memory words  
11 included in said data message are read from said  
12 changeable area;

13 time indicia for defining the time of generation of

14        said data message; and  
15  
16        memory address indicia for defining the address  
17        location in memory of the memory word included in said  
18        data message.

1        18. [Withdrawn]        The network adapter of claim 17, said  
2        send buffers further comprising:

3        a read send FIFO for storing and forwarding read  
4        request messages and response messages from said local  
5        node to said remote node;

6        a store send FIFO for storing and forwarding store  
7        messages from said local node to said remote node; and

8        an invalidation send FIFO for storing and forwarding  
9        invalidation messages from said local node to said  
10        remote node;

11        and said receive buffers further comprising:

12        a read receive FIFO for storing and forwarding read  
13        request messages and response messages from said remote

14 node to said local node;  
15 a store receive FIFO for storing and forwarding store  
16 messages from said remote node to said local node; and  
17 an invalidation receive FIFO for storing and forwarding  
18 invalidation messages from said remote node to said  
19 local node.

1 19. [Withdrawn] The network adapter of claim 18, further  
2 comprising:

3 a send FIFO selection means for prioritizing the  
4 selection of a data message from one of said three send  
5 FIFO means for transmission to said network by first  
6 selecting data messages from said invalidation send  
7 FIFO and thereafter alternatively selecting data  
8 messages from said read and store send FIFOs;

9 a receive FIFO selection means responsive to said  
10 message type differentiation indicia for selecting one  
11 of said three receive FIFO means for storing a data  
12 message received from said network; and

13 said network adapter being responsive to a node  
14 connection request for establishing a connection path  
15 to a target node, first by attempting to establish a  
16 quick connection path across a plurality of segments of  
17 said switching network to said target node, and upon  
18 determining any one of said plurality of segments is  
19 not available, issuing a camp-on connection request to  
20 said target node.

22 1 20. [Withdrawn] A memory controller for a local node of  
2 a shared memory parallel processing system, said node  
3 including a node processor, a node memory, a node cache and  
4 an I/O adapter, said system including a multi-stage  
5 switching network for communications amongst said local node  
6 and a plurality of remote nodes, said node memory including  
7 a changeable portion and an unchangeable portion; said  
8 memory controller comprising:

9 first means responsive to a request by said processor  
10 for access to a memory word for first accessing said  
11 node cache of said local node; and

12 second means responsive to said first means being



13           unable to access said memory word in said node cache  
14           for accessing said memory word selectively from a cache  
15           line in said node memory or remote memory and storing  
16           said cache line to said node cache.

1       21. [Withdrawn]     The memory controller of claim 20,  
2       further comprising:

3           remote fetch interrupt means for issuing an interrupt  
4           signal to said node processor upon determining that a  
5           requested memory word is located in remote memory for  
6           causing said node processor to switch from a first  
7           instruction stream thread to a second instruction  
8           stream thread.

1       22. [Withdrawn]     The memory controller of claim 20,  
2       further comprising:

3           data message generation means responsive to a request  
4           from a remote node for accessing a cache line  
5           identified by a remote request read address for  
6           generating a read response message to return the  
7           accessed cache line to said remote node, said read

8 response message including a message header comprising  
9 message differentiation indicia for defining said  
10 read request message type;  
11 destination node indicia equal to the sector  
12 segment of said node memory for said addressed  
13 memory word;  
14 source node indicia set to the node ID number of  
15 the local node;  
16 message length indicia for defining said read  
17 request message as being comprised of said message  
18 header only; and  
19 memory address indicia for specifying the memory  
20 address of said memory word;  
21 said data message generation means further operable for  
22 delivering said read response message to a read send  
23 FIFO of said network adapter for transmission to said  
24 network and the remote node selected by said  
25 destination node indicia.

1 23. [Withdrawn] The memory controller of claim 20,  
2 further comprising:

3 an invalidation directory;

4 cast-out means for deleting a cache line from said node  
5 cache when said cache is full to provide space for a  
6 new cache line to be stored to said cache; and for  
D2 7 sending the address of the deleted cache line to said  
8 invalidation directory to indicate said node no longer  
9 has a copy of said cache line.

1 24. [Withdrawn] The memory controller of claim 23,  
2 further comprising:

3 cast-out message generation means responsive to said  
4 cast-out means deleting a cache line addressed to a  
5 remote node for generating a cast-out message to said  
6 remote node to send the cast-out address and the local  
7 node ID number to said remote node over said network;

8 cast-out message receiving means for delivering a  
9 cast-out address and the source node ID number from the

10 message header of a cast-out message to said  
11 invalidation directory.

1 25. [Withdrawn] The memory controller of claim 20,  
2 further comprising:

3 cache copy update means for sending cache update  
4 messages to update corresponding cache lines all remote  
5 nodes having copies of a changed cache line; and

72 6 cache update message receiving means for replacing a  
7 cache line of data with an updated cache line of data  
8 received from a remote node.

1 26. [Withdrawn] The bi-directional network adapter of  
2 claim 16, said data messages further comprising:

3 a cast-out message for invalidating an invalidation  
4 directory entry at a remote node for this local node;

5 a cache copy update message for updating copies of a  
6 changed cache line at this local node at remote nodes  
7 having copies of said changed cache line; and

8 a node indicia assignment message for sending a  
9 different node number to each of the plurality of nodes  
10 of the system.

1 27. [Withdrawn] A method for operating memory controller  
2 for a local node of a shared memory parallel processing  
3 system, said node including a node processor, a node memory,  
4 a node cache and an I/O adapter, said system including a  
5 multi-stage switching network for communications amongst  
6 said local node and a plurality of remote nodes, said node  
7 memory including a changeable portion and an unchangeable  
8 portion; the method comprising the steps of:

9 responsive to a request by said processor for access to  
10 a memory word, accessing said node cache of said local  
11 node; and thereafter

12 responsive to said first means being unable to access  
13 said memory word in said node cache, accessing said  
14 memory word selectively from a cache line in said node  
15 memory or remote memory and storing said cache line to  
16 said node cache.

1 28. [Withdrawn] The method of claim 27, further

2 comprising the step of:

3 issuing an interrupt signal to said node processor upon  
4 determining that a requested memory word is located in  
5 remote memory for causing said node processor to switch  
6 from a first instruction stream thread to a second  
7 instruction stream thread.

1 29. [Withdrawn] A method for operating bi-directional  
2 network adapter for interfacing a local node of a shared  
3 memory parallel processing system to a multi-stage switching  
4 network for communications with a remote node, each said  
5 node including a node memory including a changeable portion  
6 and an unchangeable portion, and a node cache; comprising  
7 the steps of:

8 operating a plurality of send buffers for storing and  
9 forwarding data messages from said local node to said  
10 remote node over said network, and

11 operating a plurality of receive buffers for storing  
12 and forwarding a plurality of data messages from said  
13 remote node to said local node over said multi-stage  
14 network;

15 said data messages including:

16 an invalidation message for invalidating a cache  
17 line that was accessed by a remote node after said  
18 cache line has changed;

19 a read request message for requesting access of a  
20 cache line from a remote node;

21 a response message for returning a cache line over  
22 the network to a remote node that has previously  
23 requested data by a read request message; and

24 a store message storing a changed cache line to a  
25 remote node.

1 30. [Withdrawn] The method of claim 29, further  
2 comprising the steps of:

3 operating a read send FIFO for storing and forwarding  
4 read request messages and response messages from said  
5 local node to said remote node;

6 operating a store send FIFO for storing and forwarding

7 store messages from said local node to said remote  
8 node; and

9 operating an invalidation send FIFO for storing and  
10 forwarding invalidation messages from said local node  
11 to said remote node;

12 operating a read receive FIFO for storing and  
13 forwarding read request messages and response messages  
14 from said remote node to said local node;

15 operating a store receive FIFO for storing and  
16 forwarding store messages from said remote node to said  
17 local node; and

18 operating an invalidation receive FIFO for storing and  
19 forwarding invalidation messages from said remote node  
20 to said local node.

1 31. [Previously presented] A method for operating a shared  
2 memory parallel processing system as a cache coherency  
3 system including a plurality of processing nodes, each said  
4 processing node including a unique section of shared memory  
5 which is not a cache, comprising the steps of:



6 interconnecting said processing nodes through a single  
7 multi-stage communication network, said network  
8 including a dual priority switch at each node for  
9 selectively operating in normal low priority mode and  
10 camp-on high priority mode;

11 storing at each said processing node a plurality of  
12 cache lines in one or more caches;

13 distributing to each of said processing nodes a cache  
14 coherency directory;

D2  
15 tracking in said cache coherency directory which of  
16 said one or more of said processing nodes have copies  
17 of each cache line; and

18 changing said shared memory according to a two hop  
19 process including in hop 1) a requesting node  
20 requests most recent data of a home node, and in hop  
21 2) said home node immediately returns said most  
22 recent data from its shared memory to said requesting  
23 node, wherein changed data is stored immediately to  
24 said unique section of shared memory regardless of  
25 which of said nodes is changing the data and which of

26           said nodes includes the section of shared memory to  
27           be changed, wherein said shared memory always  
28           contains the most recent data.

1       32. [Previously presented] A program storage device  
2       readable by a machine, tangibly embodying a program of  
3       instructions executable by a machine to perform method steps  
4       for operating a shared memory parallel processing system  
5       including a plurality of processing nodes, each said  
6       processing node including a unique section of shared memory  
7       which is not a cache, said method steps comprising:

① 2  
8           interconnecting said processing nodes through a single  
9           multi-stage communication network, said network  
10          including a dual priority switch at each node for  
11          selectively operating in normal low priority mode and  
12          camp-on high priority mode;

13          storing at each said processing node a plurality of  
14          cache lines in one or more caches;

15          tracking in a cache coherency directory which is  
16          distributed to each of said processing nodes which of  
17          one or more of said processing nodes have copies of

18 each cache line; and

19 changing said unique section of shared memory according  
20 to a two hop process including in hop 1) a requesting  
21 node requests most recent data of a home node, and in  
22 hop 2) said home node immediately returns said most  
23 recent data from its shared memory to said requesting  
24 node, wherein changed data is stored immediately to  
25 shared memory regardless of which of said nodes is  
26 changing the data and which of said nodes includes the  
27 section of shared memory to be changed, wherein said  
28 shared memory always contains the most recent data.

1 33. [Previously presented] An article of manufacture  
2 comprising:

3 a computer useable medium having computer readable  
4 program code means embodied therein for operating a  
5 shared memory parallel processing system including a  
6 plurality of processing nodes, each said processing  
7 node including a unique section of shared memory which  
8 is not a cache, the computer readable program means in  
9 said article of manufacture comprising:

10 computer readable program code means for causing a  
11 computer to effect interconnecting said processing  
12 nodes through a multi-stage communication network, said  
13 network including a dual priority switch at each node  
14 for selectively operating in normal low priority mode  
15 and camp-on high priority mode;

16 computer readable program code means for causing a  
17 computer to effect storing at each said processing node  
18 a plurality of cache lines in one or more caches;

D<sup>2</sup>  
19 computer readable program code means for causing a  
20 computer to effect tracking in a cache coherency  
21 directory which is distributed to each of said  
22 processing nodes which of said processing nodes have  
23 copies of each cache line; and

24 computer readable program code means for storing  
25 changed data immediately to said unique section of  
26 shared memory regardless of which of said nodes is  
27 changing the data and which of said nodes includes the  
28 section of shared memory to be changed according to a  
29 two hop process including in hop 1) a requesting node  
30 requests most recent data of a home node, and in hop 2)

31           said home node immediately returns said most recent  
32           data from its shared memory to said requesting node,  
33           such that said shared memory always contains the most  
34           recent data.

1       34. [Previously presented] A computer program product or  
2       computer program element for operating a shared memory  
3       parallel processing system including a plurality of  
4       processing nodes, each said node including a unique section  
5       of shared memory which is not a cache, according to the  
6       steps of:

D2  
7           interconnecting said processing nodes through a single  
8           multi-stage communication network, said network  
9           including a dual priority switch at each node for  
10          selectively operating in normal low priority mode and  
11          camp-on high priority mode;

12          storing at each said processing node a plurality of  
13          cache lines in one or more caches;

14          distributing to each of said processing nodes a cache  
15          coherency directory;

16 tracking in said cache coherency directory which of  
17 said processing nodes have copies of each cache line;  
18 and

19 storing changed data immediately to said unique section  
20 of shared memory regardless of which of said nodes is  
21 changing the data and which of said nodes includes the  
22 section of shared memory to be changed according to a  
23 two hop process including in hop 1) a requesting node  
24 requests most recent data of a home node, and in hop 2)  
25 said home node immediately returns said most recent  
D<sup>2</sup> 26 data from its shared memory to said requesting node  
27 such that said shared memory always contains the most  
28 recent data.

1 35. [Original] The cache coherency system of claim 1,  
2 further comprising:

3 a shared memory including a first memory portion for  
4 storing unchangeable data and a second memory portion  
5 for storing changeable data; and

6 said cache coherency directory listing which nodes of  
7 said plurality of processing nodes have accessed copies

8 of said cache lines in said second memory portion.

1 36. [Original] The cache coherency system of claim 35,  
2 each of said plurality of processing nodes being operable  
3 for reading, storing, and invalidating said shared memory at  
4 any other of said processing nodes.

1 37. [Previously presented] The cache coherency system of  
2 claim 36, further comprising at a first node of said  
3 plurality of processing nodes a memory controller  
4 selectively operable first responsive to a request for  
(1)<sup>2</sup> 5 access to a memory word by first accessing the cache at  
6 said first node and, if said requested memory word is not  
7 available in said cache, selectively operable second for  
8 accessing said memory word selectively from said shared  
9 memory regardless of which of said nodes includes the  
10 section of shared memory being accessed, and storing said  
11 cache line including said memory word to said cache at said  
12 first node.

1 38. [Previously presented] The cache coherency system of  
2 claim 37, said memory controller further being selectively  
3 operable for deleting a cache line from said cache at said  
4 first node when said cache is full to provide space for a

5 new cache line to be stored to said cache, and for sending  
6 the address of the deleted cache line to an invalidation  
7 directory to indicate said node no longer has a copy of said  
8 cache line.

1 39. [Previously presented] The cache coherency system of  
2 claim 37, said memory controller further being selectively  
3 operable for sending cache update messages to update  
4 corresponding cache lines at all remote nodes having copies  
5 of a changed cache line and for receiving cache lines of  
6 data from remote nodes for updating the cache at said first  
D2 7 node.

1 40. [New] A cache coherency system for a shared memory  
2 parallel processing system including a plurality of  
3 processing nodes, comprising:

4 a multi-stage communication network for interconnecting  
5 said processing nodes;

6  
7 each said processing node including a unique section of  
8 shared memory which is not a cache;

9 each said processing node including one or more caches



10           for storing a plurality of cache lines;

11           a cache coherency directory which is distributed to

12           each of said nodes for tracking which of said nodes

13           have copies of each cache line; and

14           an adapter for storing changed data immediately to said

15           unique section of shared memory regardless of which of

16           said nodes is changing the data and which of said nodes

17           includes the section of shared memory to be changed,

18           such that said shared memory always contains the most

19           recent data.

1       41. [New] A method for operating a shared memory parallel

2       processing system as a cache coherency system including a

3       plurality of processing nodes, each said processing node

4       including a unique section of shared memory which is not a

5       cache, comprising the steps of:

6           interconnecting said processing nodes through a multi-

7           stage communication network;

8           storing at each said processing node a plurality of

9           cache lines in one or more caches;

10 distributing to each of said processing nodes a cache  
11 coherency directory;

12 tracking in said cache coherency directory which of  
13 said processing nodes have copies of each cache line;  
14 and

15 changing said shared memory, wherein changed data is  
16 stored immediately to said unique section of shared  
17 memory regardless of which of said nodes is changing  
18 the data and which of said nodes includes the section  
19 of shared memory to be changed, wherein said shared  
20 memory always contains the most recent data.

1 42. [New] A program storage device readable by a machine,  
2 tangibly embodying a program of instructions executable by a  
3 machine to perform method steps for operating a shared  
4 memory parallel processing system including a plurality of  
5 processing nodes, each said processing node including a  
6 unique section of shared memory which is not a cache, said  
7 method steps comprising:

8 interconnecting said processing nodes through a multi-  
9 stage communication network;

10 storing at each said processing node a plurality of  
11 cache lines in one or more caches;

12 tracking in a cache coherency directory which is  
13 distributed to each of said processing nodes which of  
14 said processing nodes have copies of each cache line;  
15 and

16 changing said unique section of shared memory, wherein  
17 changed data is stored immediately to shared memory  
18 regardless of which of said nodes is changing the data  
19 and which of said nodes includes the section of shared  
20 memory to be changed, wherein said shared memory always  
21 contains the most recent data.

1 43. [New] An article of manufacture comprising:

2 a computer useable medium having computer readable  
3 program code means embodied therein for operating a  
4 shared memory parallel processing system including a  
5 plurality of processing nodes, each said processing  
6 node including a unique section of shared memory which  
7 is not a cache, the computer readable program means in  
8 said article of manufacture comprising:

9 computer readable program code means for causing a  
10 computer to effect interconnecting said processing  
11 nodes through a multi-stage communication network;

12 computer readable program code means for causing a  
13 computer to effect storing at each said processing node  
14 a plurality of cache lines in one or more caches;

15 computer readable program code means for causing a  
16 computer to effect tracking in a cache coherency  
17 directory which is distributed to each of said  
D2 18 processing nodes which of said processing nodes have  
19 copies of each cache line; and

20 computer readable program code means for storing  
21 changed data immediately to said unique section of  
22 shared memory regardless of which of said nodes is  
23 changing the data and which of said nodes includes the  
24 section of shared memory to be changed such that said  
25 shared memory always contains the most recent data.

1 44. [New] A computer program product or computer program  
2 element for operating a shared memory parallel processing  
3 system including a plurality of processing nodes, each said

4 node including a unique section of shared memory which is  
5 not a cache, according to the steps of:

6 interconnecting said processing nodes through a multi-  
7 stage communication network;

8 storing at each said processing node a plurality of  
9 cache lines in one or more caches;

10 distributing to each of said processing nodes a cache  
11 coherency directory;

D2 12 tracking in said cache coherency directory which of  
13 said processing nodes have copies of each cache line;  
14 and

15 storing changed data immediately to said unique section  
16 of shared memory regardless of which of said nodes is  
17 changing the data and which of said nodes includes the  
18 section of shared memory to be changed such that said  
19 shared memory always contains the most recent data.

1 45. [New] The cache coherency system of claim 40, further  
2 comprising:

3 a shared memory including a first memory portion for  
4 storing unchangeable data and a second memory portion  
5 for storing changeable data; and

6 said cache coherency directory listing which nodes of  
7 said plurality of processing nodes have accessed copies  
8 of said cache lines in said second memory portion.

1 46. [New] The cache coherency system of claim 45, each of  
2 said plurality of processing nodes being operable for  
3 reading, storing, and invalidating said shared memory at any  
4 other of said processing nodes.

D2  
1 47. [New] The cache coherency system of claim 46, further  
2 comprising at a first node of said plurality of processing  
3 nodes a memory controller selectively operable first  
4 responsive to a request for access to a memory word by  
5 first accessing the cache at said first node and, if said  
6 requested memory word is not available in said cache,  
7 selectively operable second for accessing said memory word  
8 selectively from said shared memory regardless of which of  
9 said nodes includes the section of shared memory being  
10 accessed, and storing said cache line including said memory  
11 word to said cache at said first node.

1 48. [New] The cache coherency system of claim 47, said  
2 memory controller further being selectively operable for  
3 deleting a cache line from said cache at said first node  
4 when said cache is full to provide space for a new cache  
5 line to be stored to said cache, and for sending the address  
6 of the deleted cache line to an invalidation directory to  
7 indicate said node no longer has a copy of said cache line.

D2 1 49. [New] The cache coherency system of claim 47, said  
2 memory controller further being selectively operable for  
3 sending cache update messages to update corresponding cache  
4 lines at all remote nodes having copies of a changed cache  
5 line and for receiving cache lines of data from remote nodes  
6 for updating the cache at said first node.

---